

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI
DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION SYSTEMS

Compiler Construction (CS F363)

II Semester 2023-24

Compiler Project

Coding Details

(March 5, 2022)

Group Number 37

1. Team Members Names and IDs

ID __2021A7PS2432P__ Name_ Shreyas Kumar_____
ID __2021A7PS1449P__ Name_ Prateek Kashyap_____
ID __2021A7PS2425P__ Name_ Yuvraj Doshi_____
ID __2021A70504P__ Name_ Harshith Vasireddy_____

Mention the names of the Submitted files :

1_grammar.txt	7 parserDef.h	13 linkedList.c
2 driver.c	8 parser.c	14 stack.c
3 lexer.c	9 lexerTokenInfo.c	15 parseTree.c
4l lexer.h	10 lexerTwinBuffer.c	16 stringNTConvert.c
5 lexerDef.h	11 lexerHashTable.c	17 stringTokenConvert.c
6 parser.h	12 lexerPairLexemeToken.c	18 makefile

2. Total number of submitted files (including copy the pdf file of this coding details pro forma) : 25 (All files should be in ONE folder named as Group_#)

3. Have you compressed the folder as specified in the submission guidelines? (yes/no) Yes

4. Lexer Details:

[A].Technique used for pattern matching: longest matching algorithm using DFA

[B]. Keyword Handling Technique: __Hash function_____

[C]. Hash function description, if used for keyword handling: Simple 1D array storing a pair of lexeme and token

[D]. Have you used twin buffer? (yes/ no) _____ Yes _____

[E]. Error handling and reporting (yes/No): _____ Yes _____

[F]. Describe the errors handled by you __Lexical Errors (no matching token/long identifiers)_
Syntactical Errors (panic mode recovery)

[G]. Data Structure Description for tokenInfo (in maximum two lines): stores token value, lexeme and line number (retrieved from twinBuffer)

5. Parser Details:

[A]. High Level Data Structure Description (in maximum three lines each, avoid giving C definitions used):

- i. grammar : Array of linked lists with each linked list representing a grammar rule (and the head of the linked list being the non terminal that is being derived)
- ii. FIRST and FOLLOW sets:- Array of vector-like struct (tokenSets) used to store all the tokens in the FIRST and FOLLOW set of every non terminal
- iii. parse table :- 2D array representing the table with non terminals as rows and terminals as columns (and stores the index of the grammar rule that is to be used)
- iv. parse tree: -(Describe the node structure also):- ParseTree stores the root node of the tree (the "program" node). Every node stores an array of the children of the node (in order of insertion) and number of children

under the node, along with the symbol (terminal or non terminal), line number at which the node was inserted into the tree and the lexeme associated with the node, if any.

- v. Any other (specify and describe) :- Stack of Parse Tree Nodes, uses an underlying linked list of parse tree nodes and has a top parse tree node for O(1) retrieval

[B]. Parse tree

- i. Constructed (yes/no): Yes
- ii. Printing as per the given format (yes/no): Yes
- iii. Describe the order you have adopted for printing the parse tree nodes (in maximum two lines) :- Inorder (Leftmost Child -> Parent -> All children to the right of the leftmost child)

[C]. Grammar and Computation of First and Follow Sets

- i. Data structure for original grammar rules _Linked List_____
- ii. FIRST and FOLLOW sets computation automated (yes /no)___Yes_____
- iii. Name the functions (if automated) for computation of First and Follow sets _computerFirstAndFollow()
- iv. If computed First and Follow sets manually and represented in file/function (name that) _____

[D]. Error Handling

- v. Attempted (yes/ no): Yes
- vi. Describe the types of errors handled :- Lexical errors (no matching token according to DFA), Syntactical errors (token at top of the parse stack doesn't match input or no predicted rule)

6. Compilation Details:

- [A]. Makefile works (yes/no): Yes
- [B]. Code Compiles (yes/ no): Yes
- [C]. Mention the .c files that do not compile: _____
- [D]. Any specific function that does not compile: _____
- [E]. Ensured the compatibility of your code with the specified gcc version (yes/no) Yes

7. Driver Details: Does it take care of the options specified earlier (yes/no) Yes

8. Execution

[A]. status (describe in maximum 2 lines): Completely functioning to the best of our knowledge

[B]. Gives segmentation fault with any of the test cases (1-6) uploaded on the course page. If yes, specify the testcase file name: _____

9. Specify the language features your lexer or parser is not able to handle (in maximum one line) _____

10. Are you availing the lifeline (Yes/No):

11. Declaration: We, Shreyas, Prateek, Yuvraj and Harshith (your names) declare that we have put our genuine efforts in creating the compiler project code and have submitted the code developed only by us. We have not copied any piece of code from any source. If our code is found plagiarized in any form or degree, we understand that a disciplinary action as per the institute rules will be taken against all of us in our team and we will accept the penalty as decided by the department of Computer Science and Information Systems, BITS, Pilani.

Your names and IDs

Name: Shreyas Kumar ID: 2021A7PS2432P

Name: Prateek Kashyap ID: 2021A7PS1449P

Name: Yuvraj Doshi ID: 2021A7PS2425P

Name: Harshith Vasireddy ID: 2021A7PS0504P

Date – 5/3/2024

Not to exceed 3 pages.